

## **LISTING OF CLAIMS**

The following listing of claims is unchanged and remains pending in the present application.

### **LISTING OF CLAIMS**

1. (original) An object state transfer method in a data processing device comprising an application program, and a plurality of objects having internal states, which can be manipulated via an accessor method, the object state transfer method comprising the steps of:

arranging the internal states of the plurality of objects into a byte sequence, which is manipulated from the application program via the accessor method; and

transferring the internal states of the plurality of objects by transmitting the byte sequence to an external device.

2. (original) The object state transfer method as described in Claim 1, further comprising the steps of:

mapping an internal state of a new object to the byte sequence when the application program creates the new object; and

storing mapping data relating to the mapping.

3. (original) The object state transfer method as described in Claim 2, further comprising the step of:

when the application program has manipulated an internal state of an object by using the accessor method, setting a state in the byte sequence mapped to the internal

state or obtaining a state from the byte sequence, and returning the result to the application program.

4. (original) The object state transfer method as described in Claim 2, further comprising the step of:

transferring the byte sequence which holds the internal states of the objects and the stored mapping data to another data processing device which reproduces the objects.

5. (original) The object state transfer method as described in Claim 2, further comprising the step of:

when the application program has determined that an object becomes unnecessary, canceling the mapping between the byte sequence and the internal state of the object, and updating the mapping data.

6. (original) The object state transfer method as described in Claim 2, wherein the other data processing device which reproduces the objects performs the steps of:

receiving the byte sequence and the mapping data, transferred by using the object state transfer method as described in Claim 2;

updating a byte sequence and mapping data in the other data processing device based on the received byte sequence and mapping data, and

reproducing the objects based on the updated byte sequence and mapping data.

7. (original) An object state transfer device in a data processing device comprising an application program, and a plurality of objects having internal states, which can be manipulated via an accessor method, the object state transfer device comprising:

a region for transfer where internal states of the objects are arranged as a byte sequence;

a mapping data memory unit which stores mapping data relating to the internal states of the objects and the byte sequence; and

a transfer unit which transmits the byte sequence stored in the region for transfer and the mapping data to another data processing device.

8. (original) The object state transfer device as described in Claim 7, further comprising:

an object creation unit which, in compliance with an object creation command from the application program, performs mapping of the byte sequence and an internal state of an object to be created, stores the position of the internal state in the byte sequence into the mapping data memory unit, creates the object, and sets in the created object the position of the internal state in the byte sequence; and

a unit which, in compliance with a command to manipulate an internal state of an object from the application program via the accessor method, sets a state in the byte sequence comprising the region for transfer, or obtains a state from the byte sequence, and returns the result to the application program.

9. (original) An object state transfer device in a data processing device comprising an application program, and a plurality of objects having internal states, which can be manipulated via an accessor method, the object state transfer device comprising:

a region for transfer where internal states of the objects are arranged as a byte sequence;

a mapping data memory unit which stores mapping data relating to the internal states of the objects and the byte sequence;

a reproduction unit which receives a byte sequence representing internal states of objects transmitted from another data processing device, and mapping data relating to mapping between the internal states of the objects and the byte sequence, updates the byte sequence and the mapping data within the device itself based on the received byte sequence and mapping data, and reproduces objects having the same state as the other data processing device based on the updated byte sequence and mapping data; and

an object management unit which manages the reproduced object, and notifies the application program of data relating to the reproduced object.

10. (original) An object state transfer program which allows a computer to execute the object state transfer method as described in Claim 1.

11. (original) An object state transfer program which allows a computer to execute the object state transfer method as described in Claim 2.

12. (original) An object state transfer program which allows a computer to execute the object state transfer method as described in Claim 3.

13. (original) An object state transfer program which allows a computer to execute the object state transfer method as described in Claim 4.

14. (original) An object state transfer program which allows a computer to execute the object state transfer method as described in Claim 5.

15. (original) An object state transfer program which allows a computer to execute the object state transfer method as described in Claim 6.

16. (original) A recording medium for storing an object state transfer program which allows a computer to execute the object state transfer method as described in Claim 1.

17. (original) A recording medium for storing an object state transfer program which allows a computer to execute the object state transfer method as described in Claim 2.

18. (original) A recording medium for storing an object state transfer program which allows a computer to execute the object state transfer method as described in Claim 3.

19. (original) A recording medium for storing an object state transfer program which allows a computer to execute the object state transfer method as described in Claim 4.

20. (original) A recording medium for storing an object state transfer program which allows a computer to execute the object state transfer method as described in Claim 5.

21. (original) A recording medium for storing an object state transfer program which allows a computer to execute the object state transfer method as described in Claim 6.

22. (withdrawn) An object state transfer method in a data processing device comprising an application program, and a plurality of objects having internal states, which can be manipulated from the application program, comprising the steps of:

determining whether an object is shared among a plurality of data processing devices;

when the object is shared among the plurality of data processing devices, storing the object as a shared object in a shared heap region;

when the object is not shared among the plurality of data processing devices, storing the object in a heap region; and

transferring the states of shared objects to another data processing device by transmitting the shared heap region in unaltered form to the other data processing device.

23. (withdrawn) The object state transfer method as described in Claim 22, further comprising the steps of:

specifying beforehand a class to be transferred to the other data processing device as a transfer class;

when a command to create a new object has been issued by the application program, determining whether a class to be created, specified by the application program, is contained in the transfer class;

when the class to be created is not contained in the transfer class, creating a new object in the heap region; and

when the class to be created is contained in the transfer class, creating a new shared object in the shared heap region.

24. (withdrawn) The object state transfer method as described in Claim 22, further comprising the steps of:

determining, when the application program has issued a command to manipulate an internal state of an object, whether the object to be manipulated is stored in the shared heap region, based on an address of the object to be manipulated supplied by

the application program, and determining whether the manipulation comprises setting the internal state; and

when the object to be manipulated is stored in the shared heap region and the manipulation comprises setting the internal state, setting a flag showing that, among a plurality of blocks comprising the shared heap region, the block where the object to be manipulated is stored has been updated.

25. (withdrawn) The object state transfer method as described in Claim 22, wherein the whole of the shared heap region is transmitted to the other data processing device.

26. (withdrawn) The object state transfer method as described in Claim 24, wherein, among the plurality of blocks, only the block whose flag is set to show that its corresponding block has been updated is transmitted to the other data processing device.

27. (withdrawn) The object state transfer method as described in Claim 22, wherein another data processing device, which reproduces the shared objects, performs the steps of:

receiving the shared heap region, transferred by the object state transfer method as described in Claim 22, and arranging the transferred shared heap region to a shared heap region in the other data processing device;



calculating the difference between an address of the transferred shared heap region which is contained in the transferred shared heap region, with an address of the shared heap region in the other data processing device which is contained in the shared heap region in the other data processing device;

determining whether a pointer in a shared object contained in the transferred shared heap region points to a position in the transferred shared heap region;

correcting the pointer, based on the difference between the addresses, when the pointer points to a position in the transferred shared heap region; and

nullifying the pointer, when the pointer points to a position outside the transferred shared heap region.

28. (withdrawn) An object state transfer device in a data processing device comprising an application program, and a plurality of objects having internal states, which can be manipulated from the application program, comprising:

a shared heap region which stores, from among the plurality of objects, objects which are shared among a plurality of data processing devices, as shared objects;

a heap region which stores, from among the plurality of objects, objects which are not shared among the plurality of data processing devices; and

a transfer unit which transfers internal states of the shared objects to another data processing device by transmitting the shared heap region in unaltered form to the other data processing device.

29. (withdrawn) The object state transfer device as described in Claim 28, wherein the shared heap region comprises a plurality of blocks for storing the shared objects, and flags showing whether each of the plurality of blocks has been updated, and

wherein the object state transfer device further comprises:

a transfer class memory unit which stores a transfer class to be transferred to the other data processing device;

an object creation unit which, in response to a command from the application program to create a new object, together with the specification of a class to be created, creates a new object in the heap region in the case when the command specifies a class not contained in the transfer class as the class to be created, and creates a new shared object in the shared heap region when the command specifies a class already contained in the transfer class as the class to be created; and

a unit which, in response to a command from the application program to manipulate an internal state of an object, together with an address of the object to be manipulated, detects the command to set an internal state of an object in the shared heap region, based on the address of the object to be manipulated and the content of the manipulation, and which sets a flag which corresponds to the block where the object to be manipulated is stored, and

wherein the transfer unit transfers the whole of the shared heap region, or only the blocks whose flags are set, to the other data processing device, in compliance with a predetermined transfer method.

30. (withdrawn) An object state transfer device in a data processing device comprising an application program, and a plurality of objects having internal states, which can be manipulated from the application program, comprising:

a shared heap region which stores an address of the shared heap region itself, and objects, from among the plurality of objects, which are shared among a plurality of data processing devices, as shared objects;

a heap region which stores, from among the plurality of objects, objects which are not shared among the plurality of data processing devices;

a reproduction unit for reproducing the shared objects which receives a shared heap region transferred from another data processing device and arranges the transferred shared heap region to its own shared heap region, calculates the difference between an address of the transferred shared heap region and an address of its own shared heap region, and, among pointers in shared objects contained in the transferred shared heap region, corrects a pointer which points to a position in the transferred shared heap region, based on the difference in the addresses, and nullifies a pointer which points to a position outside the transferred shared heap region; and

an object management unit which notifies the application program of data relating to the reproduced shared object.

31. (withdrawn) An object state transfer program, which allows a computer to execute the object state transfer method as described in Claim 22.

32. (withdrawn) An object state transfer program, which allows a computer to execute the object state transfer method as described in Claim 23.

33. (withdrawn) An object state transfer program, which allows a computer to execute the object state transfer method as described in Claim 24.

34. (withdrawn) An object state transfer program, which allows a computer to execute the object state transfer method as described in Claim 25.

35. (withdrawn) An object state transfer program, which allows a computer to execute the object state transfer method as described in Claim 26.

36. (withdrawn) An object state transfer program, which allows a computer to execute the object state transfer method as described in Claim 27.

37. (withdrawn) A recording medium for storing an object state transfer program which allows a computer to execute the object state transfer method as described in Claim 22.

38. (withdrawn) A recording medium for storing an object state transfer program which allows a computer to execute the object state transfer method as described in Claim 23.

39. (withdrawn) A recording medium for storing an object state transfer program which allows a computer to execute the object state transfer method as described in Claim 24.

40. (withdrawn) A recording medium for storing an object state transfer program which allows a computer to execute the object state transfer method as described in Claim 25.

41. (withdrawn) A recording medium for storing an object state transfer program which allows a computer to execute the object state transfer method as described in Claim 26.

42. (withdrawn) A recording medium for storing an object state transfer program which allows a computer to execute the object state transfer method as described in Claim 27.